

Probabilistic Registration for Large-Scale Mobile Participatory Sensing

Sara Hachem, Animesh Pathak, Valerie Issarny
Inria Paris-Rocquencourt-France
 {sara.hachem, animesh.pathak, valerie.issarny}@inria.fr

Abstract—One of the main benefits of mobile participatory sensing becoming a reality is the increased knowledge it will provide about the real world while relying on a large number of mobile devices. Those devices can host different types of sensors incorporated in every aspect of our lives. However, given the increasing number of capable mobile devices, any participatory sensing approach should be, first and foremost, *scalable*. To address this challenge, we present an approach to decrease the participation of (sensing) devices in a manner that does not compromise the accuracy of the real-world information while increasing the efficiency of the overall system.

To reduce the number of the devices involved, we present a probabilistic registration approach, based on a realistic human mobility model, that allows devices to decide whether or not to register their sensing services depending on the probability of other, equivalent devices being present at the locations of their expected path. We present the design and implementation of a registration middleware based on our techniques, using which mobile devices can base their registration decision. Through experiments performed on real and simulated datasets, we show that our approach scales, while not sacrificing significant amounts of sensing coverage.

I. INTRODUCTION

Nowadays, most individuals carry highly advanced smart devices (e.g., phones hosting sensors, actuators, etc.), involved in every aspect of their lives. Those mobile devices can now cooperate in order to provide, among others, knowledge about the real world and perform sensing and actuating tasks, a process referred to as *participatory sensing* [1], or *mobile phones sensing* [2]. This allows regular or expert users to have more access to or insight about the real world characteristics and phenomena, be it users requesting crowd level information at a location of interest or researchers studying urban processes in a large region [3].

A participatory sensing system involves two main actors:

- 1) **Mobile Devices:** Large number of devices hosting sensors and actuators with limited energy resources.
- 2) **Registry:** Web-based component with performance capacity directly related to monetary cost.

The operation of such systems consists of two phases, for each device (shown in Figure 1):

- 1) **Registration:** The area of operation of a system can be divided into regions (e.g., cities), with a web-hosted *registry* service in charge of storing metadata (type of sensed data, API details, etc.) about each device active within that region. Each device can be assumed to periodically refresh its registration with the registry service corresponding to its current region. The need

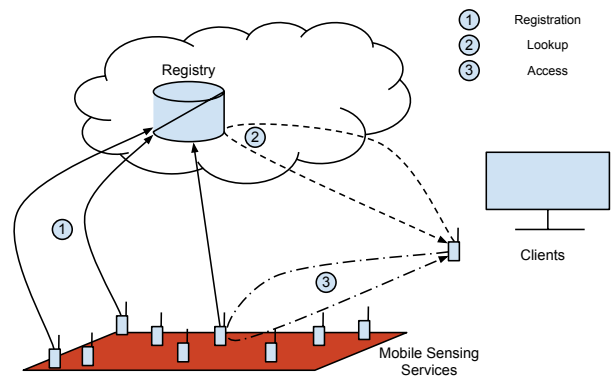


Figure 1. The operations of a mobile participatory sensing system.

for the registration process comes from the fact that participatory sensing should support any-time queries for any type of measurements and not be restricted to pre-defined sensing queries that can only be sent to near-by devices.

- 2) **Lookup and Access:** When data about the physical environment of a region is required (e.g., to display on a public screen in a city block, or by a user), the registry service is queried for devices with the relevant sensing services in its region. Those device-level services can then be queried for their data, which can then be processed to obtain desired results.

For example, consider that in the city of Paris, the system would like to provide expected crowd levels at points of interest in the city (a query such as “what is the average crowd level at the *Jardin de Tuileries* right now?”). This can be obtained by sensing the sound level surrounding the device and use it to compute the crowd level by multiplying it by an appropriate scaling factor. Needless to say, scenarios of urban mobile sensing such as the one above introduce important challenges as described below (for a detailed discussion, we refer readers to [4]):

- **Scale** - It is extremely inefficient to register and access millions of devices required to perform sensing or actuation tasks. There are several constraints to take into account, such as time, memory, and energy consumption. For example, even if half of the 12 million¹ people living in the Paris metropolitan region (and not counting the tourists) decide to register their devices, it

¹Population statistics: <http://www.recensement.insee.fr>, retrieved September 2012.

will be difficult for a single registry to process them.

- **Unknown network structure** - Mobile participatory sensing is characterized by an unknown and dynamic network structure resulting from the mobility of devices [3]. As such, applications will depend on sensing services that may not be available at the appropriate time. For instance, if an application requires the value of the wind-chill factor at a certain location, there may actually not be a wind-chill sensor at that location.



To address the above issues, especially of scale, we focus on the *registration* step. By adopting our approach, the overall system will no longer require large expensive registries with high computation and processing capabilities and can use cheaper ones with basic capacities. Furthermore, the benefits of having fewer devices register are not restricted to registration only. First, by being able to provide some level of sensing coverage of the region while reducing the number of devices that need to register, the speed of the lookup and access phases will be enhanced as well. Second, if only needed devices register, then the sensing system will not have to deal with large amounts of redundant data as long as the required coverage is satisfied. Third, the less devices get involved, the more energy will be saved till when the—now redundant—devices are needed.

Although mobile sensing coverage has been investigated before [3], the novelty of our approach for coverage estimation based on mobility of devices—previously used for resource allocations [5], routing techniques [6], [7], or aimed for robotic entities with controlled mobility [8]—is that our aim is to provide good participatory-sensing based coverage by having devices **register** their services and be **remotely accessible whenever needed** while not requiring redundant mobile sensors to operate thus involving fewer users, increasing the participatory system's performance, and saving its resources. Specifically, we make the following contributions in this paper: *i*) Mathematical formulation of the probability of a mobile phone's path being covered by registered phones providing desired functionality (Sections II and III); *ii*) Design of a middleware for probabilistic registration of mobile devices (Section V); *iii*) Evaluation of the performance of our approach (Section VI).

We then compare our work to existing solutions in Section VII and finally we present our conclusion and future work in section VIII.

II. PROBLEM DESCRIPTION

To address the high density of smart devices involved in mobile participatory sensing, the main problem we solve is whether or not a phone should register its sensors. The participatory sensing system contains several types of sensors spread over a total area of interest A . The registration decision is based on the phone's path and whether or not this path will be covered by other mobile sensors, i.e., whether other devices with similar sensing capacities will

Symbol	Meaning
Mobility	
D	Diffusion constant
v	Speed of mobile phones
t_j^f	Flight time interval of displacement j
ξ_j	Length of displacement j
t_p	Pause time
t_s	Sum of the displacement time and pause time
Deployment/Area	
A	Total area of deployment
$C_{l,r}$	Circle of center l and radius r
$\sigma_{l,r}$	Smallest square outside $C_{l,r}$ with edges parallel to the coordinate axes.
$\Sigma_{l,r}$	Largest square inside circle $C_{l,r}$ with edges parallel to the coordinate axes.
Device	
T	The set of all types of sensors
τ	A type of sensor, $\tau \in T$
N	The set of all (registered) devices
N_τ	The set of devices with sensors of type τ .
S_τ	Sensor of type τ
E_τ	Subset of T containing all sensor types that together can substitute sensor of type τ
r	The sensing range r of the device
Registration	
t_0	Time at which the new device joins the network
N_{range}^i	Number of nodes that can reach location l_i in time t_i
l_i	Location of the new device at time t_i
l_k^0	Location of node k at time t_0
δ_i^k	Displacement of node k at time t_i
d_{max}^i	Maximum distance from beyond which a device can not reach l_i in time t_i

Table I
NOTATIONS USED IN THIS PAPER.

be present at this device's future locations when it passes through them. The path is represented as a sequence of (x, y) coordinates and timestamps representing the times at which the device will reach each pair of coordinates. Notations are summarized in Table 1. Descriptions of the network and device characteristics on which we build our solution follow.

A. Network Characteristics

The mobile network has the following characteristics:

- **Large number of sensors:** The network consists of a large set N of mobile phones with each embedding several types of sensors. N_τ is a subset of N containing sensors of type τ , and T is the set of possible types of sensors. A sensor is denoted as S_τ , where $\tau \in T$.
- **Dynamic structure:** The most commonly used devices in mobile participatory sensing are smart phones hosting several types of sensors. Each mobile device κ moves to different locations l_i^κ , starting from l_0^κ and ending at l_{final}^κ . As mentioned earlier, the participatory sensing system should operate on a global scale and support dynamic queries. As such, it is not possible to assume that on-the-fly queries (that do not involve registries) with no prior knowledge of this dynamic network will be appropriate. The reason is that such an approach will require the involvement of a large number

of devices for message and data routing purposes, a requirement that is common to traditional mobile sensing solutions [8], [9] and goes against our objective to decrease the number of involved devices.

B. Mobile Device Characteristics

A mobile device possesses the following characteristics:

- **Location-awareness:** Each device is equipped with the ability to identify its location as (x, y) coordinates (e.g., GPS receiver) [3].
- **Path awareness:** Each device in our system is aware of the path it will follow and the final destination to reach during a mobility period. While future paths can be predicted for routine trips of people, for the less-predictable paths, one can rely on the fact that it is increasingly common for individuals using vehicles to follow paths specified by the navigation systems on their devices, a trend also increasing among pedestrians.² We acknowledge the possibility that some users might not have navigation systems on their devices. In such case, it is also possible for users to provide their future locations as input (e.g., tourists providing a list of landmarks they plan on visiting at future times).
- **Mobility:** All devices in the network are mobile. Devices following the same mobility model (pedestrian, in vehicles, etc.) have, on average, a constant speed v (an assumption commonly made in existing solutions [10]).
- **Fixed, identical sensing range:** All sensors in the network have an identical sensing range r and each sensor has a 360° coverage and can sense events in a circle $C_{l_i, r}$ with radius r and centered at location l_i . This is known as the boolean disk model [11].

C. Problem Formulation

Based on the network and mobile node characteristics presented above, we formulate the problem to solve as follows:

Given a network with an unknown dynamic topology, consisting of a set N of location-aware mobile devices hosting T different types of sensors such that there are N_τ sensors of each type τ , determine the probability P_{cov} that the locations on the path to be followed by a new mobile device with a given itinerary and sensor type can be covered by equivalent nodes in N .

Each device computes P_{cov} locally with the help of the Registry. The latter provides the device with the number of already registered devices and their distribution in space³. This allows us to have each device use the knowledge of

its path locally without having to store this information on the registry as the knowledge of the devices' distribution in space, instead of the estimated path of each registered device, allows us to protect users' privacy and save storage space. It should be noted that the device and the registry exchange at most 2 messages. The first message contains the number of registered devices and the type of their distribution in space along with the values of its parameters (e.g., $\{n: 5000, \text{dist: Normal}(0, 50)\}$). The second message, the registration message, is sent only if the device decides to register its services. Therefore, our approach does not introduce a large overhead on the communication cost with the increasing number of devices, especially that the size of the messages is relatively small.

III. PROBABILISTIC REGISTRATION

Our approach to handle the problems arising from the large scale of mobile participatory sensing systems, in the registration phase, is founded on using probabilistic mobility models to estimate the locations of mobile devices in a large network. The resulting knowledge can then be exploited by an incoming device to make a decision whether or not to register its services based on an estimate of the probability that any registered device with similar capabilities will cross paths with it. For this purpose, we choose the Truncated Lévy Walk mobility model [7], which we recall below.

A. Mobility Model: Truncated Lévy Walk

We choose the **Truncated Lévy Walk (TLW)** as our mobility estimation model, which has been shown in recent works, to best represent the mobility of humans (pedestrian, vehicles, etc.). TLW assumes that entities have a constant speed depending on the mobility category they belong to (pedestrians, vehicles, bicycles, etc.)⁴ [6]. In this model, entities select a uniformly distributed direction and a Lévy distributed length of displacement, after which they chose a Lévy distributed pause period where they stay in the same location. The total displacement period is the sum of their actual movement period and the pause period. To summarize, TLW has the following characteristics:

- **Motion speed:** Motion speed v is constant.
- **Time interval of each displacement:** The time interval of each displacement depends on the velocity and length of the displacement.
- **Direction of the displacement:** The direction angle θ_i is uniformly distributed in $[0, 2\pi)$.
- **Length of the displacement:** The length of the displacement is assumed to have a Lévy distribution. The range of the displacement length varies between $(0, \tau_\xi)$, where τ_ξ is the maximum allowed displacement length. More details can be found in [7], [12], [13].

²Services such as <http://www.google.com/mobile/maps/> provide turn-by-turn driving and walking navigation.

³For registered nodes, it is possible to estimate their spatial distribution along with its parameters, given their location, using curve fitting techniques (such as those provided by Matlab's ALLFITDIST method).

⁴For simplification, we only consider devices in the same category.

- **Pause interval:** TLW includes a pause time, that takes place after the end of each displacement where devices stay at the same location. The pause time should not exceed a pre-defined value τ_p which is the maximum allowed pause interval.
- **Complete displacement time:** This time, t_s , is the sum of the displacement time and the pause time.

B. Probabilistic Registration Approach

We aim with our **probabilistic registration** to allow only a subset of mobile devices to register their sensors based on the network sensing coverage. We consider that the coverage requirement (threshold) depends on the sensor and can be specified in a knowledge base describing metadata about sensors and actuators (more details can be found in [14]).

As shown in Figure 2, the future path of a mobile device is a sequence of locations $(l_1, \dots, l_i, \dots, l_{final})$ which the device will pass through. Such information in practice can be obtained by the personal navigation system of the mobile device. Without loss of generality, we assume that the length of any segment (l_i, l_{i+1}) is less than 2 times the sensing range r , thus allowing us to treat only the vertices as locations of interest while ensuring coverage.

1) *Atomic Coverage:* As mentioned earlier, we need to compute the probability that the path of a new device hosting a sensor of type τ is covered. For this purpose, the approach should compute the probability that at least one device will be at each of the locations of interest at the same time as the new device. In this section, we assume that N contains only sensors of type τ . Let l_i be the location of the device after time period t_i , and $P(\geq 1 \text{ device } \kappa \text{ is at } (l_i, t_i))$ be the probability we are looking for at each location.

$$P(\geq 1 \text{ device } \kappa \text{ at } (l_i, t_i)) = 1 - P(\text{no device at } (l_i, t_i))$$

$$\text{Where } P(\text{no device at } (l_i, t_i)) = \prod_{\kappa \in N} (1 - P(\kappa \text{ at } (l_i, t_i)))$$

The probability that device κ will be at location l_i at time t_i is the probability of device κ moving from its initial location to l_i . This refers to the total displacement δ_i^κ of device κ until time t_i from its initial location l_κ^0 at $t = 0$. The probability to compute becomes:

$$P(\geq 1 \text{ device } \kappa \text{ at } (l_i, t_i)) = 1 - \prod_{\kappa \in N} (1 - (P(\delta_i^\kappa) = l_i - l_\kappa^0))$$

Using [6], we define the probability that device κ is in $C_{l_i, r}$ (to approximate l_i to an area instead of a point) at time t_i starting from a known location $l^0 = (X_\kappa^0, Y_\kappa^0)$ as

$$\begin{aligned} f(l_i, t_i, r, l^0) &= P(C_{l_i, r} \text{ has a device at } t_i) \\ &= \oint_{l_\kappa \in C_{l_i, r}} \phi(X_\kappa, Y_\kappa, t_i) dl_\kappa \\ &= \frac{1}{2\pi Dt_i} \oint_{l_\kappa \in C_{l_i, r}} e^{\frac{(X_\kappa - X_\kappa^0)^2 + (Y_\kappa - Y_\kappa^0)^2}{2Dt_i}} dl_\kappa \end{aligned} \quad (1)$$

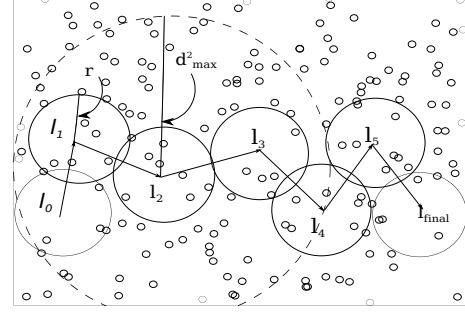


Figure 2. The big circle C_{l_i, d_{max}^i} limits the range of devices that can reach l_2 in time t_2 while the smaller circles showing the approximate location at which sensors should be to consider that the location is covered.

Where, D is the diffusion factor in the Truncated Lévy Walk and it is equal to $\frac{\sigma_\xi^2}{\mu_t}$ [13], where σ_ξ^2 is the variance of the displacement length and μ_t is the mean of the complete displacement time distribution. We consider σ_ξ^2 and μ_t to be parameters that depend on the real life scenario and the mobility category; for example, pedestrians do not have the same σ_ξ^2 and μ_t values as vehicles.

Since we do not keep track of all locations of already registered devices, the initial location l_κ^0 for device κ at time t_0 can be any coordinate in the deployment area A . Going a step further, we can say that with respect to location l_i , we only care for devices located within the circle C_{l_i, d_{max}^i} with center l_i and radius d_{max}^i ; d_{max}^i is the distance from l_i beyond which no device κ can start from l_κ^0 at t_0 and reach location l_i at time t_i . Taking into account all locations within circle C_{l_i, d_{max}^i} , the probability of device κ moving to $C_{l_i, r}$ becomes⁵:

$$\begin{aligned} g(l_i, t_i, r) &= \oint_{l_\kappa^0 \in C_{l_i, d_{max}^i}} PDF(X_\kappa^0) * PDF(Y_\kappa^0) dl_\kappa^0 \\ &\quad * \frac{1}{2\pi Dt_i} \oint_{l_\kappa \in C_{l_i, r}} (e^{\frac{(X_\kappa - X_\kappa^0)^2 + (Y_\kappa - Y_\kappa^0)^2}{2Dt_i}} dl_\kappa) \end{aligned}$$

If we go back to the probability of at least one device being at location l_i , we obtain by substitution:

$$P(\geq 1 \text{ device at } (l_i, t_i)) = 1 - \prod_{\kappa \in N} (1 - g(l_i, t_i, r))$$

Now that we have determined the probability of coverage at one location, we can repeat the process to obtain the complete probability of coverage, i.e. having at least one device at each location, with L being the total number of locations:

$$P_{cov} = \prod_{l_i \in L} (1 - \prod_{\kappa \in N} (1 - g(l_i, t_i, r))) \quad (2)$$

2) *Coverage By Composition:* If the locations of interest on the device's path are not covered by sensors of its type τ , the device can check if its sensor can be substituted by a composition of other types of sensors. In fact, we have

⁵Note that X_κ^0 and Y_κ^0 are independent.

developed an ontology presented in [14] that determines what types of sensors can compose their measurements to substitute other types. Based on this ontology, the probability of coverage can also be computed in terms of those substitute types. Let E_τ be a set containing substitute types that together can replace the missing sensor. Let's suppose for example that a wind-chill sensor (type τ) can be substituted by a thermometer (type τ_1) and an anemometer (type τ_2), we now need to compute the probability that all locations l_i on the path of the new device will be covered by both sensor types τ_1 and τ_2 . We have already showed how to compute P_{cov} for the atomic case, i.e. where no composition takes place. In the following we will show how we can compute the probability that takes substitute sensors into account.

let $P_{cov}^{S_\tau}$ be the probability of coverage by sensor type τ , and SP_{cov}^τ be the probability of coverage by all types in E_τ .

$$\begin{aligned} SP_{cov}^\tau &= P_{cov}(\text{ by } S_{\tau_1} \text{ and } S_{\tau_2}) \\ &= P_{cov}^{S_{\tau_1}} * P_{cov}^{S_{\tau_2}} \end{aligned}$$

In the general case:

$$SP_{cov}^\tau = \prod_{S \in E_\tau} P_{cov}^S$$

P_{cov}^S is computed using the same equation as P_{cov} for the atomic case, with a change in the type of the sensor to evaluate (Eq. 2).

The complete probability of coverage, i.e., the probability including both atomic coverage and coverage with composition cases, becomes:

$$\begin{aligned} P_{cov} &= P_{cov}(\text{ by } S_\tau \text{ or } E_\tau) \\ &= 1 - (P_{cov}^{S_\tau} * SP_{cov}^\tau) \\ &= 1 - ((1 - P_{cov}^{S_\tau}) * (1 - SP_{cov}^\tau)) \end{aligned}$$

In a more general form, assuming there is a set J of possible substitution sets, E_j , the complete probability of coverage is:

$$P_{cov} = 1 - ((1 - P_{cov}^{S_\tau}) * \prod_{E_j \in J} (1 - SP_{cov}^{E_j}))$$

C. Computation-Related Simplifications

As stated earlier, for each location of interest on the device's path, it suffices to consider N_{range}^i devices, which is the set of devices that can actually reach l_i on time. To compute the value of N_{range}^i , we apply the following steps:

- 1) For each location l_i , compute $d_{max}^i = v * (t_i - t_0)$.
- 2) For each location l_i , compute for the following values:
 $x_{min} = x_i - d_{max}^i$ and $x_{max} = x_i + d_{max}^i$ for X_κ and
 $y_{min} = y_i - d_{max}^i$ and $y_{max} = y_i + d_{max}^i$ for Y_κ . Those values will provide us with boundaries (for squares) that can be used to determine the areas for each location beyond which locations are not accessible in time t_i .

Using those limits, and based on the fact that the distribution of locations of devices is provided by the registry when the new device joins the network, we compute the probability that $l_i = (X_i, Y_i)$ has values between the upper and lower limits we just computed, i.e. $P(X_i \in [x_{min}, x_{max}])$ and $P(Y_i \in [y_{min}, y_{max}])$. Then, we use the resulting probability, to compute the number of devices expected to be within the defined boundaries based on the following product:

$$N_{range}^i = P(X_i \in [x_{min}, x_{max}]) * P(Y_i \in [y_{min}, y_{max}]) * N$$

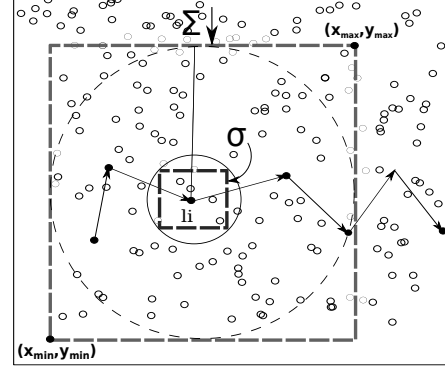


Figure 3. A closer look at the substitution of $C_{l_i, r}$ by $\sigma_{l_i, r}$ and C_{l_i, d_{max}^i} by Σ_{l_i, d_{max}^i} .

Moreover, to simplify the complex integration computations, we replace the circle $C_{l_i, r}$ by the largest square $\Sigma_{l_i, r}$ within $C_{l_i, r}$. This allows us to split the area integrals into double integrals. Further, we let σ_{l_i, d_{max}^i} be the smallest square outside a circle C_{l_i, d_{max}^i} (Figure 3). σ_{l_i, d_{max}^i} is chosen to be a square for the same reason we chose the square $\Sigma_{l_i, r}$ instead of $C_{l_i, r}$ to limit the integral bounds. The above simplifications reduce P_{cov} equation to:

$$\begin{aligned} P_{cov} \simeq & \prod_{l_i \in L} (1 - \prod_{\kappa \in N_{range}^i} (1 - \oint_{l_\kappa^0 \in \Sigma_{l_i, r}} PDF(X_\kappa^0) \\ & * PDF(Y_\kappa^0) dX_\kappa^0 dY_\kappa^0 * \frac{1}{2\pi Dt_i} \\ & \oint_{l_\kappa \in \sigma_{l_i, d_{max}^i}} e^{\frac{(X_\kappa - X_\kappa^0)^2 + (Y_\kappa - Y_\kappa^0)^2}{2Dt_i}} dX_\kappa dY_\kappa) \end{aligned} \quad (3)$$

We can separate the X-axis integrals from the Y-axis integrals, and the equation becomes:

$$\begin{aligned} P_{cov} \simeq & \prod_{l_i \in L} (1 - \prod_{\kappa \in N_{range}^i} (1 - \frac{1}{2\pi Dt_i} * \\ & \Phi(X_i, X_\kappa, X_\kappa^0) * \Phi(Y_i, Y_\kappa, Y_\kappa^0))) \end{aligned}$$

Where

$$\Phi(a, b, c) = \int_{a-d_{max}^i}^{a+d_{max}^i} \int_{b-\frac{\sqrt{2}*r_s}{2}}^{b+\frac{\sqrt{2}*r_s}{2}} PDF(c) * e^{\frac{(b-c)^2}{2Dt_i}} dbdc$$

The same simplification logic applies to coverage by composition since it also builds on Eq. 2.

D. Support for Mobility Models

We presented above a mathematical solution that is specific to TLW. However, it is easy to plugin any other mobility model as long as it provides a formula to estimate the probability of a mobile device being at location l_i at time t_i . Precisely, the model to plugin, should provide an equation that computes the same result as Eq. 1. Consequently, the new equation will be used to reconstruct Eq. 3 following the same steps we presented above. The steps are defined in the knowledge base and retrieved to be computed dynamically whenever a new model is to be plugged in. It should be noted that although we provide the possibility for new models to be introduced, our approach is based on TLW and we have not evaluated the effects of replacing it with other models. Evaluations of the newly introduced models are to be performed by the developer.

IV. LOOK-UP & ACCESS

Although the paper addresses the registration of devices only, we find it important to briefly describe our work on look-up and access in order to give a full picture of our global solution. Our registration middleware is one of two components of a *discovery* middleware, the second being a *look-up* middleware. The look-up middleware handles the search for and selection of services that provide the required functionality. It retrieves the addresses of such services and returns them to an *access* middleware. The latter communicates with the services and acquires their measurements. A brute force approach for look-up entails the selection of **all** appropriate registered services. However, to further decrease the number of active devices we are currently investigating a probabilistic look-up approach that returns only a subset of appropriate services based on their distribution in space and the type of the event to sense. The size of the subset to select should be determined dynamically based on a tradeoff between communication and resource consumption costs and the quality of coverage.

V. PROBABILISTIC REGISTRATION MIDDLEWARE

As mentioned in the introduction, there are two main actors in our probabilistic registration approach: the **mobile device** and the **Registry**. The former hosts the registration middleware and all computations are local; the latter is accessed by devices to register their services (Figure 4).

A. Architecture

In addition to maintaining information on the registered services, the Registry provides the new mobile device with information on the network density along with the spatial distribution of mobile devices hosting registered services.

As for the registration middleware, it comprises three components: i) a substitution sets generator which determines possible compositions, ii) a probability estimator that

computes the probability of coverage, iii) a registration estimator that determines whether or not the new device should register its services based on the computed probability.

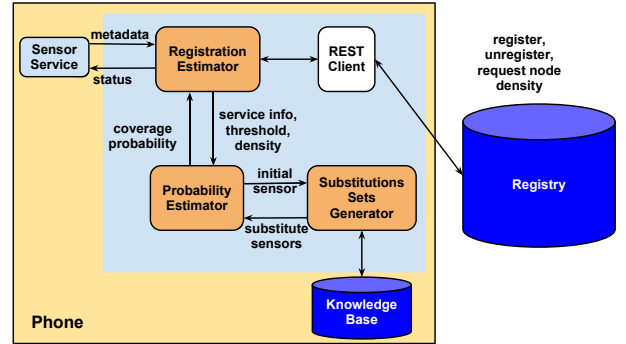


Figure 4. Architecture of the Registration Middleware and Registry.

B. Implementation

We implemented our middleware using Java 1.6. The Registry is developed as a RESTful Web service and for the back-end storage storage we use the Apache Derby database. Our middleware is deployed on Android devices and laptops, and for the knowledge base we use Androjena⁶. Our code is available at <http://forge.ow2.org/projects/choreos/>.

The Registry provides an API to enable the registration of mobile devices with two public methods:

- `findNumberOfServices()`: The Registry keeps track of the number of all registered sensor services (per sensor type) and provides this information on request.
- `registerService()`: The Registry stores metadata of services in a back-end database for a certain period of time that is specified by the hosting device.

The registration middleware, hosted on mobile devices, provides an API to compute the coverage probability and generate the registration decision with two public methods:

- `generateRegistrationDecision()`: This method enables the middleware to generate the final registration decision based on the coverage requirements, and the computed coverage probability.
- `registerService()`: This method enables the registration middleware to send the service metadata to the Registry if the registration decision is true. Each service is registered for a certain period, after which the registration decision should be re-evaluated and the Registry should be informed whether or not to keep it.

In our current implementation, the complex mathematical computation is performed on a (cloud-hosted, stateless) web service, invoked by the device. However we intend to explore the latency and performance tradeoff between this option and performing the complex mathematical computations locally on the device as part of our future work. The emergence

⁶Androjena: <http://code.google.com/p/androjena>.

of applications performing similar tasks on smartphones⁷ is a positive development towards that end. In our trials, discussed next, we observed that the actual computation time was independent of the number of nodes, with a mean of 3.084 ms and standard deviation of 1.4 ms.

VI. EVALUATION

A. Evaluation Based on Real Traces

To include the large scale and real world aspects in our evaluation, we use a dataset that provides real mobility traces — in the form of (*taxi id, date time, longitude, latitude*)— for a large number of GPS-equipped taxis in Beijing [15] which we considered as our mobiles devices. We assume that a device that registers provides sensing services until the end of the simulation. We focused on an area of length 1136.2 km and breadth 3002.4 km.

To be able to utilize this data, we sampled the traces to simulate a Poisson arrival process and then computed the average velocity of the devices (8 km/hr) and set their sensing range to be 10 m. Finally, based on techniques from [6], we computed the diffusion value; the result is 44.

B. Results

We evaluate the correctness and performance of our registration approach (through the implemented registration middleware) in terms of two criteria. Firstly, we evaluate how *coverage* varies as we shift from deterministic to probabilistic registration, i.e., as the coverage requirement decreases. We define *Maximum Possible Coverage* (MPC) as the maximum percentage of the area of interest that can be covered by registered mobile devices. We first computed the MPC for all 10000 devices (the taxis in the downloaded trace) in the deployment area. The resulting MPC is 0.003% of the total area. We consider the result to be low, but were unable to find larger datasets to evaluate our approach with higher densities.

The results show that our approach satisfies the threshold requirements for real mobility traces. As we can see in Figure 5(a), a threshold of 0.8 results in 80% of the total coverage, and a threshold of 0.6 results in 60% of the total coverage. However, given that the MPC is too low, devices will keep registering as they show up, which explains the increasing registration percentage in Figure 5(a).

To clearly see the scalability effect, it is important to reach a MPC of 100% before all 10000 devices register. However, instead of adding phantom traces to increase the maximum possible coverage of the set of devices — which would have taken away the value of real mobility traces— we decided to restrict the area of focus and increase the sensing range. Since computational geometry-based techniques to obtain an optimal configuration for this

would have been too time consuming, and our goal was only to find a suitable scenario with the above properties, we decided to graphically and rapidly render the possible options, and chose the first to match our constraints by visual confirmation. Results showed that a sensing range of 10 kilometers best addresses the sparsity issue between devices in this data set spread over a very large area. We are aware that this modification is not an accurate reflection of the real world, but in the absence of better publicly available large scale mobility traces, it was necessary to evaluate if our probabilistic registration approach decreases the number of registered devices while maintaining the requested coverage threshold; in fact, larger ranges may indeed be applicable for sensing, say, air quality. Figure 5(b) illustrates how our approach successfully addresses the large scale issue by preventing devices from registering (the number of devices that register for a threshold of 0.8 is less than 2000) while still achieving the required coverage. In conclusion, as the number of registered devices increases, our approach allows nodes to register if need be (Figure 5(a)) and prevents them from doing so otherwise (Figure 5(b)) while satisfying the required coverage in both cases.

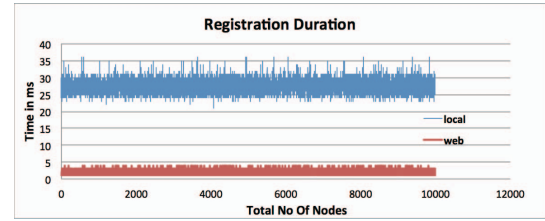


Figure 6. The time needed for 10000 devices to register with threshold = 1, local represents the time needed to generate the registration decision and web represents the time needed to register services in the Registry.

Secondly, to evaluate the *time needed for a service to register* as the density of registered devices increases, we registered all the 10000 devices sequentially with threshold equal to 1. We only evaluate the time needed for services to register for this threshold because our aim is to compute the upper limit for registration time, and lower thresholds will surely lead to faster registration times as the density of registered devices will be less. We divided the registration time into two phases: the first phase, *local registration*, is the time needed for the Registration middleware to determine whether or not the service should register. The second phase, *web registration* is the time needed for the Registry Manager to actually register the service. The results (Figure 6) show us that registration times remain between 2 and 4 ms and between 25 and 30 ms for first and second phases consecutively.

VII. RELATED WORK

In this section, we discuss three categories of existing works that relate to our approach: registration of devices, mobile coverage, and mobile participatory sensing.

⁷Mathomatic: <https://play.google.com/store/apps/details?id=com.mathomatic&hl=en>.

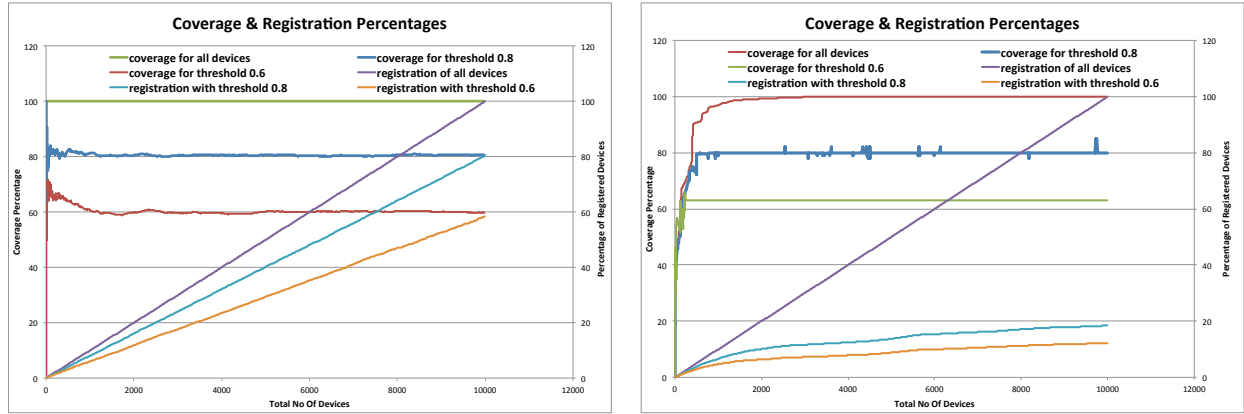


Figure 5. The coverage and registration percentages as the threshold decreases from 1 to 0.6 a) for a radius of 0.005 km b) for a radius of 10 km.

A. Registration of Devices

Existing solutions for registering devices as services adopt traditional discovery techniques where **all** devices register their information and their service metadata in order to be accessible, such as the work in [16]–[18]. Hence, devices register their services without taking the characteristics of the network and the density of registered services into account. As such, the available solutions do not provide any alternative to filter out some of the available services once they all satisfy the request criteria.

Further, in many existing works related to Wireless Sensor Networks, devices are grouped in small local networks, and they discover and communicate with each other directly [16], [17]. The direct relation among sensors is not applicable in our case for many reasons, including that the request for information does not necessarily come from the same region where devices providing sensing services are, and the devices are spread over large areas, and are thus not able to discover and communicate with each other directly to determine which of them should register. Several works in the WSN area have tried to decrease the number of active devices by adopting duty cycling or device selection techniques where some devices provide services while others sleep. However, we consider that their solutions apply more to the look-up stage of the discovery process and not the registration stage because, even though devices do not provide services at one time or another, they are still known to the network and to their neighbors [10]. Further, the main goal of those solutions is to decrease energy consumption and not increase the overall system performance.

B. Probabilistic coverage

To the best of our knowledge, there are not many works that utilize probabilistic coverage in participatory sensing. Among existing solutions, authors in [10] model pedestrians as mobile sensor devices moving according to a discrete Markov chain. This mobility model simplifies the motion of pedestrians and is not accurate enough to present human mobility, especially as compared to more recent models

such as the truncated Lévy Walk model we adopted in our work. Authors in [6] also study the mobility of humans to compute coverage. However, the aim of their work is to have a better likelihood to deliver messages based on the users' proximity to the final destination, which is computed based on the estimation of a device's *diffusion*⁸ and distance from the destination. Similarly, in [7] authors study the mobility of humans but for routing purposes. Both works presented above do not provide the solutions to the challenges (Section I) that we address.

Other existing solutions consider mobile devices to be robotic entities. In [8] the mobility of sensors is addressed to better collect information carried in mobile devices by properly placed sink devices. They consider two mobility models: *Random Walk* model and *Guass-Markov* model, which we consider unfit in our case to solve mobile participatory sensing challenges. In [19] entities have a controlled mobility based on a specified model, with the aim of providing a coverage map. Each sensor collects the position information from all other sensors to store it in an information table and exchange it with others, which is not feasible in our case.

C. Participatory sensing

Existing participatory sensing solutions take the scale issue into account from two perspectives:

- *User scale* when a large number of smart phone carrying users are involved in the participatory sensing process [20], [21]; and
- *data scale* when the sensing process generates large amounts of data that should be categorized, filtered out or condensed [21], [22]

However, to the best of our knowledge, those solutions address both issues at the data level and not at the user device level. As such all willing devices are asked for their measurements which might result in large amounts of redundant data.

⁸For more details on diffusion, we refer readers to [6], [7].

VIII. CONCLUSION AND FUTURE WORK

We have presented in the paper a probabilistic registration approach that addresses the large scale issue of smart devices involved in mobile participatory sensing. The approach relies on the fact that paths of mobile devices in dense networks are bound to cross and as such devices can substitute one another based on their services. Our approach checks if coverage can take place by substituting a device with a similar one or by a composition of the functionality of other devices hosting other types of sensors. To do so, we compute the probability that the substitution sensors hosted on mobile devices can cover the path of the new device. We implemented and evaluated the correctness of our approach and results show that our middleware successfully decreases the number of participating devices without decreasing the sensing coverage beyond an acceptable limit.

As for our future work, we plan on investigating learning techniques to predict the mobility of humans instead of requiring a priori knowledge of their future paths. Further, the probabilistic registration middleware is to be integrated in a discovery middleware that can also handle look-up of mobile devices, defined as the process of searching for and selecting registered services based on their attributes. Continuing our work, we aim at developing a probabilistic look-up approach that handles the large scale issue by returning only a subset of appropriate registered services.

This work is partially supported by the European Commission's Seventh Framework Programme FP7/2007-2013 under grant agreement number 257178 (project CHOREOS - <http://www.choreos.eu>), and the ANR MURPHY project. We thank professor Qin Lv, our shepherd, for her insight and help in preparing the camera-ready version of this paper.

REFERENCES

- [1] E. Paulos, R. Honicky, and E. Goodman, "Sensing atmosphere," *Human-Computer Interaction Institute*, p. 203, 2007.
- [2] N. Lane, E. Miluzzo, H. Lu, D. Peebles, T. Choudhury, and A. Campbell, "A survey of mobile phone sensing," *Communications Magazine, IEEE*, vol. 48, no. 9, 2010.
- [3] D. Raychaudhuri and M. Gerla, *Emerging Wireless Technologies and the Future Mobile Internet*. Cambridge University Press, 2011.
- [4] V. Issarny, N. Georgantas, S. Hachem, A. Zarras, P. Vassiliadis, M. Autili, M. Gerosa, and A. Hamida, "Service-oriented middleware for the future internet: state of the art and research directions," *Journal of Internet Services and Applications*, vol. 2, no. 1, pp. 23–45, 2011.
- [5] J. Krumm and E. Horvitz, "Predestination: Inferring destinations from partial trajectories," *UbiComp 2006: Ubiquitous Computing*, pp. 243–260, 2006.
- [6] U. Sadiq and M. Kumar, "Proximol: Proximity and mobility estimation for efficient forwarding in opportunistic networks," in *2011 Eighth IEEE International Conference on Mobile Ad-Hoc and Sensor Systems*. IEEE, 2011, pp. 312–321.
- [7] I. Rhee, M. Shin, S. Hong, K. Lee, S. Kim, and S. Chong, "On the levy-walk nature of human mobility," *IEEE/ACM Transactions on Networking (TON)*, vol. 19, no. 3, 2011.
- [8] G. Keung, Q. Zhang, and B. Li, "The delay-constrained information coverage problem in mobile sensor networks: single hop case," *Wireless Networks*, vol. 16, no. 7, 2010.
- [9] S. Awwad, N. Noordin, and M. Rasid, "Coverage degree in mobile nodes wireless sensor network," in *Telecommunication Technologies 2008 and 2008 2nd Malaysia Conference on Photonics. NCTT-MCP 2008. 6th National Conference on*. IEEE, 2008, pp. 273–277.
- [10] A. Ahmed, K. Yasumoto, Y. Yamauchi, and M. Ito, "Distance and time based node selection for probabilistic coverage in people-centric sensing," in *Proc. 8th Annual IEEE Intl. Conf. Sensor, Mesh and Ad Hoc Communications and Networks (SECON-2011)*, 2011, pp. 134–142.
- [11] B. Wang, "Sensor coverage model," *Coverage Control in Sensor Networks*, pp. 19–34, 2010.
- [12] S. Kim, C. Lee *et al.*, "Superdiffusive behavior of mobile nodes and its impact on routing protocol performance," *Mobile Computing, IEEE Transactions on*, vol. 9, no. 2, pp. 288–304, 2010.
- [13] I. Rhee, M. Shin, S. Hong, K. Lee, and S. Chong, "On the levy-walk nature of human mobility," in *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*, april 2008, pp. 924–932.
- [14] S. Hachem, T. Teixeira, and V. Issarny, "Ontologies for the Internet of Things," Proceedings of the PhD Student Symposium of the ACM/IFIP/USENIX 12th International Middleware Conference. [Online]. Available: <http://dl.acm.org/citation.cfm?id=2093193>
- [15] J. Yuan, Y. Zhen, X. Xie, and G. Sun, "Driving with knowledge from the physical world," in *The 17th ACM SIGKDD international conference on Knowledge Discovery and Data mining, KDD'11*, 2011.
- [16] S. Karnouskos, D. Savio, P. Spiess, D. Guinard, V. Trifa, and O. Baecker, "Real-world service interaction with enterprise systems in dynamic manufacturing environments," *Artificial Intelligence Techniques for Networked Manufacturing Enterprises Management*, pp. 423–457, 2010.
- [17] D. Guinard, V. Trifa, S. Karnouskos, P. Spiess, and D. Savio, "Interacting with the SOA-Based internet of things: Discovery, query, selection, and on-demand provisioning of Web Services," *IEEE transactions on Services Computing*, vol. 3, no. 3, pp. 223–235, 2010.
- [18] V. Tsiatsis, A. Gluhak, T. Bauge *et al.*, "The SENSEI real world internet architecture," 2010.
- [19] J. Luo and Q. Zhang, "Probabilistic coverage map for mobile sensor networks," in *Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008. IEEE*. IEEE, 2008, pp. 1–5.
- [20] M. Von Kaenel, P. Sommer, and R. Wattenhofer, "Ikarus: large-scale participatory sensing at high altitudes," in *Proceedings of the 12th Workshop on Mobile Computing Systems and Applications*. ACM, 2011, pp. 63–68.
- [21] H. Lu, W. Pan, N. Lane, T. Choudhury, and A. Campbell, "Soundsense: scalable sound sensing for people-centric applications on mobile phones," in *Proceedings of the 7th international conference on Mobile systems, applications, and services*. New York, NY, USA, 2009, pp. 165–178.
- [22] M. Mun, S. Reddy, K. Shilton, N. Yau, J. Burke, D. Estrin, M. Hansen, E. Howard, R. West, and P. Boda, "Peir, the personal environmental impact report, as a platform for participatory sensing systems research," in *Proceedings of the 7th international conference on Mobile systems, applications, and services*, ser. MobiSys '09, 2009, pp. 55–68.